

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторної роботи

«Створення баз даних за допомогою мови Transact SQL»

з курсу «Проектування баз даних» для студентів
спеціальності 122 Комп'ютерні науки, з курсу «Бази даних»
спеціальності 186 Видавництво та поліграфія та з курсу «Проектування
інформаційних систем» спеціальності 124 Системний аналіз

Затверджено
редакційно-видавничою
радою університету,
протокол № 1 від 30.01.2018 року

Харків
НТУ «ХПІ»
2018

Методичні вказівки до лабораторної роботи «Створення баз даних за допомогою мови Transact SQL» з курсу «Проектування баз даних» для студентів спеціальності 122 Комп'ютерні науки та інформаційні технології, з курсу «Бази даних» спеціальності 186 Видавництво та поліграфічна справа та з курсу «Проектування інформаційних систем» спеціальності 124 Системний аналіз / уклад.: Л. Б. Кашеєв, С. В. Коваленко, О. С. Куценко – Харків : НТУ «ХПІ», 2018. – 16 с.

Укладачі: Л. Б. Кашеєв,
 С. В. Коваленко,
 О. С. Куценко

Рецензент Л. М. Любчик

Кафедра системного аналізу та інформаційно-аналітичних технологій

ВСТУП

Метою даної лабораторної роботи є ознайомлення з можливостями мови Transact-SQL для створення реляційних баз даних (БД), таблиць, що нормалізовані до третьої нормальної форми, побудови зв'язків між таблицями. Необхідно розглянути способи перевірки коректності введення даних, як за допомогою простих умов, так і використовуючи призначені для користувача функції. База даних повинна зберігати інформацію деякої предметної області, що передбачає введення конкретних даних до таблиць.

1. ТЕОРЕТИЧНІ ОСНОВИ

Система управління базами даних (СУБД) – це комп'ютеризована система структурованих даних, основна мета якої – зберігання інформації та надання її на вимогу користувача.

У реляційній базі даних ми будемо виділяють наступні елементи:

1. Дані, які зазвичай називають постійними, хоча вони звичайно не є такими в загальноприйнятому розумінні. Коли ця інформація збережена в таблицях, вона стає частиною постійних даних або тягне за собою зміни постійних даних.

2. Об'єкти. У реляційних БД це таблиці (інша назва – відношення), що описують деякі об'єкти реального світу.

3. Зв'язки, які відображають залежності між об'єктами. Як правило, вони бувають двосторонніми.

4. Властивості. Всі об'єкти і зв'язки мають певні властивості. Властивості об'єктів визначаються полями таблиці. Властивості зв'язків виражаються в їх характеристиках при формуванні.

Рейтинг систем управління базами даних, випущений аналітичним агентством Теглайн в 2016 році і сформований на основі анкетування, помістив Microsoft SQL Server на третє місце серед усіх відомих.

Microsoft SQL Server – це система управління клієнт-серверними реляційними базами даних, орієнтована на роботу під управлінням операційних систем Microsoft Windows. MS SQL Server реалізується у вигляді кількох самостійних служб, кожна з яких відповідає за виконання певних завдань. При цьому виділяють серверну і клієнтську

частини.

Утиліта SQL Server Management Studio дуже зручна у використанні і надає наступні можливості: управління настройками MS SQL Server; конфігурація системи безпеки (управління ролями, обліковими записами, віддаленими серверами); робота зі структурою баз даних (створення, редагування і видалення самої БД та її елементів); управління виконанням завдань; демонстрація поточної активності та багато іншого.

В MS SQL Server у кожного поля таблиці, локальної змінної, виразу і параметра є визначений тип даних. Тип даних являє собою атрибут, який визначає, якого роду дані можуть зберігатися в об'єкті: цілі числа, символи, дані грошового типу, мітки часу і дати, двійкові рядки і так далі. MS SQL Server (мова Transact-SQL) надає набір системних типів даних, що визначають всі типи даних, які можуть використовуватися в ньому (табл. 1.1).

Таблиця 1.1 – Типи даних, що використовуються в мові Transact-SQL

Назва	Розмір	Діапазон значень, призначення
Цілі числа:		
Bit	1 байт	змінна (поле таблиці) зберігається в 1 байте, якщо таких значень до 8, то вони розміщуються в тому ж байті
Bigint	8 байт	64-розрядне ціле число, дозволяє зберігати числа від -2^{63} до $+2^{63}-1$
Int	4 байти	від $-2\,147\,483\,648$ до $+2\,147\,483\,647$
SmallInt	2 байти	від -32768 до $+32767$
TinyInt	1 байт	від 0 до 255
Числа з фіксованою комою:		
Decimal (Numeric)		від $-10^{38}-1$ до $+10^{38}-1$
Money	8 байт	грошовий формат, діапазон значень від -2^{63} до $+2^{63}$ з чотирма знаками після коми
SmallMoney	4 байти	грошовий формат, діапазон значень від $-214748,3648$ до $+214748,3647$

Назва	Розмір	Діапазон значень, призначення
Числа з плаваючою комою:		
Float		від $-1,79E+308$ до $+1,79E+308$
Real	4 байти	від $-3,40E+8$ до $-1,18E+38$
Дата та час:		
DateTime	8 байт	від 1 січня 1753 року по 31 грудня 9999 року з точністю до 0,02 секунди
DateTime2	6-8 байт	підтримує точність до 0,1 мкс
SmallDateTime	4 байти	від 1 січня 1900 року до 6 червня 2079 року з точністю одна хвилина
DateTimeOffset	8-10 байт	аналогічний типу DateTime, але зберігає зсув відносно часу UTC
Date	3 байти	дата від 1 січня 0001 року по 31 грудня 9999 року
Time	3-5 байт	час з точністю до 0,1 мс
Символьні рядки:		
Char		рядок фіксованої довжини. Максимальна довжина рядка 8000 символів
VarChar		рядок змінної довжини. Максимальна довжина рядка 8000, але при використанні ключового слова «max» може зберігати до 2^{31} байт
NChar		рядок фіксованої довжини в Unicode. Максимальна довжина рядка 4000 символів
NVarChar		рядок змінної довжини в Unicode. Максимальна довжина рядка 4000 символів, але при використанні ключового слова «max» може зберігати до 2^{31} байт

2. РОЗРОБКА СТРУКТУРИ БАЗИ ДАНИХ

В якості предметної області, що досліджується, зупинимось на добре відомій всім студентам інформації про успішність навчання в

рамках академічної групи. Щоб не перевантажувати матеріал лабораторної роботи поставимо обмеження – кількість таблиць, що є основою бази даних, не повинно перевищувати трьох.

У цьому випадку доцільно виділити 3 сутності – Студенти, Дисципліни та Іспити.

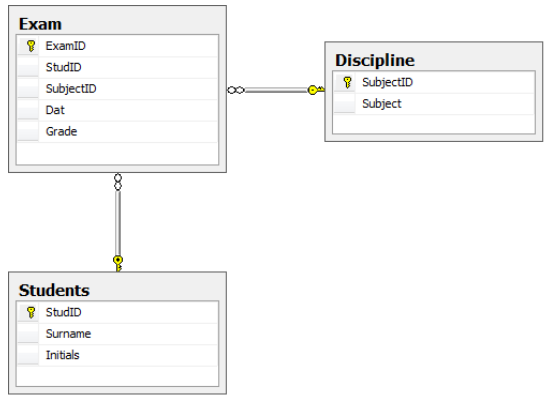


Рисунок 2.1 – Інфологічна модель бази даних, що розробляється

Структура бази даних (рис. 2.1) представлена трьома зв’язаними таблицями, дві з яких є батьківськими (*Students* і *Discipline*), одна дочірня (*Exam*). Побудовано дві відносини між таблицями, що носять назву «один-до-багатьох», при цьому таблиці *Students* і *Discipline* виступають на стороні «один», тобто для них повинні бути визначені первинні ключі.

Подібне відношення між таблицями *Students* і *Discipline* можна розглядати в більш широкому сенсі, виділивши їх зв’язок як «багато-до-багатьох», який в свою чергу реалізується через проміжну таблицю *Exam*.

Таблиця 2.1 – Структура таблиці *Students*

Назва поля	Тип	Розмір	Призначення
StudID	Int	4	Унікальний ідентифікатор
Surname	NVarChar	25	Прізвище студента
Initials	NVarChar	4	Ініціали студента

Таблиця 2.2 – Структура таблиці Discipline

Назва поля	Тип	Розмір	Призначення
SubjectID	Int	4	Унікальний ідентифікатор
Subject	NVarChar	30	Назва дисципліни

Таблиця 2.3 – Структура таблиці Exam

Назва поля	Тип	Розмір	Призначення
ExamID	Int	4	Унікальний ідентифікатор
StudID	Int	4	Код студента
SubjectID	Int	4	Код дисципліни
Dat	Date	3	Дата складання іспиту
Grade	TinyInt	1	Оцінка

3. ПОСТАНОВА ЗАДАЧІ

У даній лабораторній роботі пропонується створити в середовищі SQL Server Management Studio за допомогою мови Transact-SQL програму, яка буде створювати базу даних, що складається з трьох зв'язаних таблиць Students, Discipline та Exam (зазначеної вище структури). Для таблиць задані деякі обмеження:

1. У таблиці Students прізвище студента повинне складатися тільки з букв українського алфавіту, ніякі інші символи неприпустимі.

2. У таблиці Students ініціали повинні бути визначені як послідовність, що складається з української літери, точки, української літери, точки.

3. У таблиці Exam оцінка, що отримана з іспиту, повинна знаходитися в діапазоні від 1 до 100 балів.

Після створення структури таблиць необхідно заповнити їх даними, використовуючи команди мови Transact-SQL.

4. ПОЧАТКОВІ НАЛАШТУВАННЯ ДЛЯ РОБОТИ З БАЗОЮ ДАНИХ

Програмний код, що розробляється, передбачає багаторазове використання, модифікацію і запуск. Тому першою дією перевіряємо

чи міститься інформація про нашу БД Data в системній таблиці sys.databases і якщо відповідь позитивна, переводимо її в режим одного користувача та видаляємо з диска. Наступним кроком заново створюємо базу даних Data та робимо її активною.

```
use master
go
if exists (Select * From sys.databases Where
name='Data')
begin
    Alter Database Data set single_user with rollback
    immediate
    Drop Database Data
end
go
Create Database Data
go
use Data
go
```

Закінчуватися програмний код повинен переведенням створеної бази даних в багатокористувацький режим, з можливістю її подальшої обробки засобами утиліти SQL Server Management Studio.

```
Alter Database Data set multi_user
go
```

5. СТВОРЕННЯ КОРИСТУВАЦЬКОЇ ФУНКЦІЇ

Як обмеження на введення даних в полі прізвища студента (таблиця Students), необхідно використовувати тільки букви українського алфавіту. Якби довжина прізвища була фіксована, то можна було б скористатися функцією Like, яка дозволяє використовувати перевірку за шаблоном (перевірку за шаблоном реалізуємо для ініціалів). У нашому випадку напишемо окрему функцію перевірки, що призначена для користувача, – вона повинна розташовуватися в програмі до першого свого виклику. В якості

формального параметра функції будемо передавати рядок, що аналізується. Основний алгоритм полягає в циклічній перевірці всіх символів рядка на потрапляння в інтервал українських букв. Як результат функція повинна повернути логічне значення true/false допустимості використання набору символів в якості прізвища студента:

```
Create Function dbo.F (@f nvarchar(25))
Returns bit
AS
Begin
    Declare @i int=1, @b bit='True'
    while @i<=len(@f)
    begin
        if lower(substring(@f,@i,1)) not between 'a'
        and 'я'
            set @b='False'
        set @i=@i+1
    end
    Return @b
End
go
```

6. СТВОРЕННЯ ЗВ'ЯЗАНИХ ТАБЛИЦЬ

Основною частиною лабораторної роботи є створення таблиць. Необхідно пам'ятати, що в першу чергу створюються батьківські таблиці, а тільки потім дочірні. Використання команди Create Table вимагає для кожного поля таблиці зазначення імені, типу, можливості приймати невизначені значення, правило введення та значення за замовчуванням. Таблиці Students і Discipline обов'язково повинні мати первинні ключі для зв'язку з дочірньою таблицею.

```
Create Table Students(
    StudID Int Not Null Identity(1,1) Primary Key,
    Surname Nvarchar(25) Not Null Default 'Шевченко'
```

```

Check (dbo.F(Surname)='True'),
Initials Nvarchar(4) Not null Default 'A.A.',
Check (Initials Like '[A-Я].[A-Я].')
)

Create Table Discipline(
    SubjectID Int Identity(1,1) Not Null Primary Key,
    Subject Nvarchar(30) Not Null
)

```

Таблиця Exam повинна містити два зовнішніх ключа, за допомогою яких буде здійснено зв'язок з батьківськими таблицями.

```

Create Table Exam(
    ExamID Int Not Null Identity(1,1) Primary Key,
    StudID Int Not Null,
    SubjectID Int Not Null,
    Dat Date Null,
    Grade Tinyint Not Null Default 49 Check (Grade
    between 1 and 100),
    Foreign Key (StudID) References Students
    (StudID),
    Foreign Key (SubjectID) references Discipline
    (SubjectID)
)

```

7. ЗАПОВНЕННЯ ТАБЛИЦЬ ДАНИМИ

У мові SQL є можливість програмного заповнення таблиць інформацією. Призначена для цього команда Insert Into має кілька синтаксисів. Скористаємося найпростішим з них, в якому дані вводяться у вигляді конкретних значень. Зв'язки між таблицями припускають допустимість введення в дочірню таблицю тільки значень, що містяться в батьківських таблицях. Отже, спочатку заповнюємо батьківські таблиці і тільки потім дочірню.

```
Insert Into Students (Surname, Initials) Values  
( 'Борисов', 'Т.В.' )
```

```
Insert Into Students (Surname, Initials) Values  
( 'Петров', 'В.Я.' )
```

```
Insert Into Students (Surname, Initials) Values  
( 'Охрименко', 'А.К.' )
```

```
Insert Into Discipline (Subject) Values  
( 'Програмування' )
```

```
Insert Into Discipline (Subject) Values  
( 'Комп'ютерна графіка' )
```

```
Insert Into Discipline (Subject) Values ( 'Фізика' )
```

```
Insert Into Discipline (Subject) Values ( 'Бази  
даних' )
```

```
Insert Into Exam (StudID, SubjectID, Dat, Grade)  
Values (1,2, '2017-01-21', 94)
```

```
Insert Into Exam (StudID, SubjectID, Dat, Grade)  
Values (2,2, '2016-12-28', 75)
```

```
Insert Into Exam (StudID, SubjectID, Dat, Grade)  
Values (3,1, '2017-01-15', (Select max(Grade) From  
Exam))
```

```
Insert Into Exam (StudID, SubjectID, Dat, Grade)  
Values (1,3, '2017-01-21', 63)
```

```
Insert Into Exam (StudID, SubjectID, Dat, Grade)  
Values (2,4, '2016-12-28', 82)
```

Для перегляду підсумків виконаної лабораторної роботи можна скористатися командою Select, результат роботи якої представлений на рис. 7.1.

```
SELECT ExamID, Surname, Initials, Subject, Dat,  
Grade  
FROM Discipline INNER JOIN Exam  
ON Discipline.SubjectID = Exam.SubjectID  
INNER JOIN Students ON Exam.StudID =  
Students.StudID
```

Results		Messages				
	ExamID	Surname	Initials	Subject	Dat	Grade
1	1	Борисов	Т.В.	Комп'ютерна графіка	2017-01-21	94
2	2	Петров	В.Я.	Комп'ютерна графіка	2016-12-28	75
3	3	Охрименко	А.К.	Програмування	2017-01-15	94
4	4	Борисов	Т.В.	Фізика	2017-01-21	63
5	5	Петров	В.Я.	Бази даних	2016-12-28	82

Рисунок 7.1 – Результат виведення даних з трьох таблиць

8. ВАРІАНТИ ЗАВДАНЬ НА ЛАБОРАТОРНУ РОБОТУ

Варіант 1


За допомогою мови SQL виконати наступні дії:

Створити базу даних, що містить три зв'язані таблиці. Перша таблиця містить поля – код і назва факультету. Друга таблиця – код факультету, код спеціальності, найменування спеціальності. Третя таблиця – номер залікової книжки, прізвище, ініціали, код спеціальності, стать, дату народження. Для таблиць визначити правила введення даних (в разі необхідності скористатися функціями):

1. Назва факультету повинна складатися з одного або трьох слів.
 2. Найменування спеціальності складається тільки з українських літер.
 3. Номер залікової книжки представлено у форматі – дві літери, тире, 5 цифр.
 4. Студент повинен бути не молодше 17 років.
- Ввести 10-12 записів в три таблиці.

Варіант 2

За допомогою мови SQL виконати наступні дії:

Створити базу даних, що містить три зв'язані таблиці. Перша таблиця містить поля – код і найменування фірм виробників автомобілів. Друга таблиця – код фірми-виробника автомобілів, код моделі автомобіля, найменування моделі авто, довжина авто (мм), ширина авто (мм). Третя таблиця – державний номер  автомобіля, код моделі автомобіля, тип двигуна, колір. Для таблиць визначити правила введення даних (в разі необхідності скористатися функціями):

1. Назва фірми-виробника складається з великих українських

літер або будь-яких латинських.

2. Назва моделі автомобіля обов'язково включає цифру.
3. Державний номер автомобіля починається на «АХ», потім йдуть 4 цифри, потім дві українські літери.
4. Колір автомобіля можна вибирати тільки зі списку 7 кольорів веселки.

Ввести 10-12 записів в три таблиці.

Варіант 3

За допомогою мови SQL виконати наступні дії:

Створити базу даних, що містить три зв'язані таблиці. Перша таблиця містить поля – код кольору, колір. Друга таблиця – код сімейства, найменування сімейства тварин (наприклад, котяті, бегемотові, ведмежі, слонові і т. п.). Третя таблиця – порядковий номер, найменування тварини, кличка тварини, код сімейства, код кольору, вага, здатність плавати. Для таблиць визначити правила введення даних (в разі необхідності скористатися функціями):

1. Колір можна вибрати тільки зі списку 7 кольорів веселки, білого та чорного кольорів.
2. Назва сімейства складається з одного слова і закінчується на голосну літеру.

3. Ціла частина ваги тварини містить цифру 0.

4. Кличка тварини складається з українських літер і пробілів.

Ввести 10-12 записів в три таблиці.

Варіант 4

За допомогою мови SQL виконати наступні дії:

Створити базу даних, що містить три зв'язані таблиці. Перша таблиця містить поля – код, прізвище абонента. Друга таблиця – код, найменування вулиці. Третя таблиця – номер телефону, код прізвища, ініціали, індекс, код вулиці, номер будинку, номер квартири, стать абонента. Для таблиць визначити правила введення даних (в разі необхідності скористатися функціями):

1. Прізвище абонента складається з українських літер, при цьому починається з великої літери.

2. Найменування вулиці складається не більше ніж з 3 слів.

3. Номер будинку складається з цифр, може містити в середині

«/» і закінчуватися на літеру. Наприклад, 34, 7/8, 78a

4. Номер телефону в форматі 999-99-99 або 99-99-99.

Ввести 10-12 записів в три таблиці.

Варіант 5

За допомогою мови SQL виконати наступні дії:

Створити базу даних, що містить три зв'язані таблиці. Перша таблиця містить поля – код, найменування видавництва. Друга таблиця – код автора, прізвище автора, ініціали, дата народження. Третя таблиця – код книги, код прізвища автора, найменування книги, код видавництва, кількість сторінок, жанр книги. Для таблиць визначити правила введення даних (в разі необхідності скористатися функціями):

1. Назва видавництва складається з одного або двох слів.

2. Дата народження письменника не пізніше – 1 січня 1990 року.

3. Прізвище автора записується тільки латинськими або українськими літерами (при цьому вони не можуть перемішуватися).

4. Кількість сторінок в книзі є парним числом.

Ввести 10-12 записів в три таблиці.

Варіант 6

За допомогою мови SQL виконати наступні дії:

Створити базу даних, що містить три зв'язані таблиці. Перша таблиця містить поля – код фільму, найменування фільму, тривалість. Друга таблиця – код каналу, назва телевізійного каналу. Третя таблиця – код, код найменування фільму, код каналу, час початку, час закінчення, обмеження за віком. Для таблиць визначити правила введення даних (в разі необхідності скористатися функціями):

1. Тривалість фільму лежить в межах від 70 до 120 хвилин.

2. Найменування телевізійного каналу складається тільки з українських літер і цифр або тільки з латинських літер і цифр (K1, Інтер, Fox, Футбол1).

3. Час початку фільму менше ніж час закінчення на 90 і більше хвилин.

4. Фільми з обмеженням за віком більше 18 років можуть демонструватися тільки після 22.00.

Ввести 10-12 записів в три таблиці.

Варіант 7

За допомогою мови SQL виконати наступні дії:

Створити базу даних, що містить три зв'язані таблиці. Перша таблиця містить поля – код і найменування країни. Друга таблиця – код країни, код фірми-виробника годинників, найменування фірми виробника годинників, дата заснування фірми. Третя таблиця – код годинника, код фірми-виробника годинників, назва моделі годинника, дата випуску, призначення (чоловічі, жіночі, дитячі тощо), вартість. Для таблиць визначити правила введення даних (в разі необхідності скористатися функціями):

1. Назва фірми-виробника складається з латинських літер та пробілів.
 2. Дата заснування фірми-виробника годинників після 1510 року.
 3. Вартість годинників (окрім дитячих)– позитивне число, що не перевищує 10000 грн.
 4. Вартість дитячих годинників не перевищує 1000 грн.
- Ввести 10-12 записів в три таблиці.

Контрольні запитання

1. Дайте визначення понять база даних і таблиця?
2. Назвіть основні типи даних, які використовуються в Transact-SQL?
3. Чим відрізняються рядкові типи NChar і NVarChar?
4. У чому полягає особливість використання типу Bit?
5. Проведіть аналіз числових типів з фіксованою і плаваючою комами.
6. Яка команда використовується для видалення бази даних?
7. Які команди використовуються для створення бази даних і таблиць?
8. Назвіть типи відносин між двома таблицями. Поясніть їх особливості, наведіть приклади.
9. Що означає параметр not null в описі поля таблиці, що створюється?
10. Навіщо потрібен первинний ключ (Primary Key)?
11. Як задається обмеження на введення даних в полі таблиці?
12. Дайте пояснення поняття зовнішній ключ таблиці (Foreign

Key). Яке його основне призначення?

13. Яким чином в мові SQL відбувається додавання даних в таблицю?

СПИСОК ЛІТЕРАТУРИ

1. Кренке Д. Теория и практика построения баз. СПб.: Питер, 2005. 864 с.

2. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. Москва: Вильямс, 2000. 1120 с.

3. Ролланд Ф. Д. Основные концепции баз данных. Москва: Вильямс, 2002. 256 с.

4. Грофф Дж. Р., Вайнберг П. Н., Оппель Э. Дж. SQL: полное руководство. Москва: Вильямс, 2015. 959 с.

5. Кашеев Л. Б., Коваленко С. В. Методичні вказівки до лабораторних робіт з курсу «Бази даних та інформаційні системи» для студентів напряму 6.040302 «Інформатика». – Харків : НТУ «ХПІ», 2015. 56 с.